

## Chapter 3 : Operator

Oleh : Fiftin Noviyanto

### A. Apa Definisi Operator?

- Operator adalah aksi yang digunakan untuk memproses variabel atau angka. Contoh operator untuk memproses angka, antara lain : penambahan (+), Pengurangan (-), Perkalian (\*), dan Pembagian (/).
- Setiap simbol-simbol tersebut merupakan contoh operator yang memproses angka dan menghasilkan suatu jawaban tertentu
- Contoh :
  - 5 + 6 menghasilkan 11
  - 13 \* 5 menghasilkan 65
  - 12/2 menghasilkan 6

- Tidak semua operator mengembalikan nilai hasil penghitungan, seperti operator perhitungan ( +, -, \*, dan / ). Terdapat operator yang menghasilkan nilai true atau false (bukan angka).

- Contoh, penggunaan operator kurang dari (<) dan lebih dari (>):

15 < 37      15 kurang dari 37  
menghasilkan nilai true

14 > 100      14 lebih dari 100  
menghasilkan nilai false

- Akibat menggunakan operator, seperti +, -, \* (multiplication), and / (division), diperlukan proses komputasi untuk memprosesnya.
- Diperlukan pemahaman terhadap proses komputasi terhadap kinerja mesin. Seperti urutan prioritas operator (pemahaman matematika).

#### Urutan Operasi

- Tidak semua operator memiliki prioritas yang sama.
- Beberapa operator memiliki prioritas lebih penting dari operator lainnya.

- Operator yang memiliki operator paling tinggi akan dieksekusi terlebih dahulu oleh komputer.
- Operator lainnya akan dieksekusi berdasarkan urutan prioritasnya.
- Contoh : Apabila pada suatu ekspresi terdapat operator penambahan, pengurangan, perkalian, maka perkalian memiliki prioritas lebih tinggi, sehingga akan dieksekusi terlebih dahulu.
- Contoh :
 
$$3 + 2 * 5$$
- Pada kasus di atas, maka  $2 * 5$  akan dieksekusi terlebih dahulu, hasil perkaliannya tersebut dijumlahkan dengan 3, sehingga menghasilkan nilai 13

- Contoh :
 

Ranking operators	1st	2nd
	↓	↓
	4 * 6	- 3
Becomes	↓	↓
	24	- 3
Which then becomes	↓	↓
	21	
- Pada contoh ini,  $4 * 6$  dihitung terlebih dahulu. Selanjutnya, 3 dikurangi dengan 24 sehingga menghasilkan 21.
- | Ranking operators  | 1st   | 2nd |
|--------------------|-------|-----|
|                    | ↓     | ↓   |
|                    | 8 / 2 | - 5 |
| Becomes            | ↓     | ↓   |
|                    | 4     | - 5 |
| Which then becomes | ↓     | ↓   |
|                    | -1    |     |
- Pada contoh ini,  $8/2$  dihitung terlebih dahulu. selanjutnya, 5 digunakan untuk mengurangi 4 sehingga menghasilkan -1.

- Urutan Priority Operator  
PEMDAS = PIPO LONDO
- PEMDAS = parentheses (dalam kurung), exponents, perkalian, pembagian, penambahan dan pengurangan
- Contoh :
 

	8 (5 + 2) - 7
	2 <sup>nd</sup> 1 <sup>st</sup> 3 <sup>rd</sup>
Becomes	8 (7) - 7
	56 - 7
	49
- Dikarenakan terdapat dalam kurung, maka (5+2) dieksekusi terlebih dahulu. selanjutnya, 8 dikalikan dengan 7 menghasilkan 56. Kemudian 7 digunakan untuk mengurangi 56 sehingga menghasilkan 49

## B. Operator Binary dan Unary

**Operator Binary**

- Operator binary adalah operator yang memerlukan dua operand. Antara dua operand tersebut membutuhkan operator.
- Contoh :
 
$$6 + 7 \text{ atau } 3 * 5, \text{ dll}$$

**Operator Unary**

- Operator Unary adalah operator yang hanya membutuhkan satu operand. Tanda negative (-) merupakan salah satu contoh operator Unary.
- Contoh :
 
$$- 4$$
 Operand 4. hasilnya adalah - 4.

- Operator Unary dan Binary

$$-(8 + 7)$$

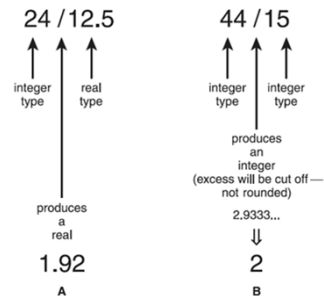
- Operator + adalah operator binary. Sedangkan operandnya adalah 8 dan 7. Operator Unary adalah tanda negative. Statement  $8 + 7$  dieksekusi terlebih dahulu, selanjutnya hasilnya diberikan operator unary, sehingga nilainya -15.

## C. Operator Aritmatika

- Setiap bahasa pemrograman menggunakan operator aritmatika untuk melakukan proses perhitungan.
- Operator aritmatika akan dibaca seperti halnya proses pada kalkulator.

+	penambahan
-	pengurangan
*	perkalian
/	pembagian

## Pembagian: a Special Case



### Contoh

- $6/4$  menghasilkan 1 instead of 1.5  
(0.5 is thrown away)
- $8/4$  menghasilkan 2
- $-143/7$  menghasilkan -20  
(0.429 is thrown away)

## D. Operator Relational

- Terdapat operator yang menghasilkan nilai dari hasil perbandingan antara beberapa nilai.
- Contoh perbandingan antara dua nilai dengan operasi : lebih besar, lebih kecil, sama dengan.
- Hasil dari operator relational adalah true / false.
- Berikut ini adalah operator-operator relational :  
 $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ,  $==$ .
- Pada operator terakhir terdapat double sama dengan ( $==$ ). Operator tersebut berbeda dengan tanda ( $=$ ). ( $==$ ) memiliki arti sama nilai dengan.

### • Contoh :

Operation	Result
15 < 16	True
15 is less than 16	
-13 > 100	False
-13 is greater than 100	
25 == 25	True
25 is equal to 25	
35 == 4	False
35 is equal to 4	

Operation	Result
57 <= 69	True
57 is less than or equal to 69	
12.5 >= 12.5	True
12.5 is greater than or equal to 12.5	
-26 <= -27	False
-26 is less than or equal to -27	

Operation	Result
28 != 28	False
28 is not equal to 28	
0 != 7	True
0 is not equal to 7	

## E. Operator Logic

- Operator logic adalah kelompok operator yang memungkinkan komputer menghasilkan keputusan kompleks.
- Penghubungnya disebut dengan operator logika.
- Contoh :  
**Contoh di Real-world** : Andi rajin membaca and Andi selalu mendapat nilai A  
**Contoh Programming** : x lebih besar dari 24 dan x kurang dari 30  
 $x > 24$  and  $x < 30$   
**Real-world example**: Andi akan menikah atau Andi mendaftar kuliah  
**Programming example**: y kurang dari 0 atau y lebih besar dari 100  
 $y < 0$  or  $y > 100$

- Pada konteks pemrograman, hasil dari operator aritmatika bukan berupa hasil angka, namun nilai true/ false.

### Contoh dengan Variables

**jumlah > 12 or jumlah < 0**

Bagaimana jika **jumlah** bernilai 14?

Bagaimana jika **jumlah** bernilai 25?

Bagaimana jika **jumlah** bernilai 12?

- Operator logika **AND** akan menghasilkan nilai TRUE apabila kedua ekspresi bernilai TRUE. Sedangkan operator logika **OR** akan menghasilkan nilai TRUE apabila salah satu ekspresi bernilai TRUE.
- Operator AND, menggunakan symbol double ampersands: **&&**. Sedangkan operator OR menggunakan symbol dua garis vertical: **||**.
- Contoh :

<code>x &gt; 3 and x &lt; 10</code>	is the same as	<code>x &lt; 3 &amp;&amp; x &lt; 10</code>
<code>y &gt; 100 or y &lt; 0</code>	is the same as	<code>y &gt; 100    y &lt; 0</code>

## F. Operator logika khusus: Operator NOT

- Operator not berada di depan statement. Ini merupakan operator Unary karena hanya membutuhkan satu operand.
- Operator NOT menggunakan symbol tanga seru(!)
- Contoh :

<code>not (x &lt; 3)</code>	sama dengan	<code>!(x &lt; 3)</code>
<code>not (y &gt; 100)</code>	sama dengan	<code>!(y &gt; 100)</code>

Expression	Result
<code>sum = 14;</code>	
<code>!(sum &gt; 12)</code>	
<code>14 &gt; 12</code>	
↓	
<code>true</code>	
<code>not true</code>	<code>false</code>

## G. A Powerful Operator untuk bahasa pemrograman: Mod

- Operator MOD, ketika digunakan pada dua operand akan menghasilkan sisa hasil bagi.
- Operator MOD direpresentasikan dengan simbol persen (%).

- Contoh :

28	<i>mod</i>	14	is	0	because there is no remainder.
172	<i>mod</i>	35	is	32	because $172 \div 35 = 4$ with a remainder of 32.
1943	<i>mod</i>	7	is	4	because $1943 \div 7 = 277$ with a remainder of 4.
18	<i>mod</i>	17	is	1	because $18 \div 17 = 1$ with a remainder of 1.

## Simpulan

- Pada chapter ini telah dibahas tentang operator dan penggunaannya terhadap angka maupun variable.
- Operator digolongkan sebagai binary and unary yang menunjukkan jumlah operand yang dibutuhkan.
- Operator lainnya : arithmetic, relational, and logic operators.